



TITLE:

先行順序付き合流可能運搬経路問題に対する局所探索法 (最適化手法の深化と広がり)

AUTHOR(S):

吉良, 知文; 岩根, 秀直

CITATION:

吉良, 知文 ...[et al]. 先行順序付き合流可能運搬経路問題に対する局所探索法 (最適化手法の深化と広がり). 数理解析研究所講究録 2012, 1773: 77-86

ISSUE DATE:

2012-01

URL:

<http://hdl.handle.net/2433/171712>

RIGHT:

先行順序付き合流可能運搬経路問題 に対する局所探索法

九州大学大学院数理学府 吉良 知文 (Akifumi Kira)
Graduate School of Mathematics, Kyushu University
(株) 富士通研究所 岩根 秀直 (Hidenao Iwane)
FUJITSU LABORATORIES LTD.

1 はじめに

小売店への商品の配送, ゴミ収集車や除雪車の運行経路など, 複数の運搬車 (vehicle) を用いて顧客 (customer) に需要を運搬/収集するときの最適なルートを求める問題は様々なバリエーションが考えられており, これらを総称して運搬経路問題, あるいは配送計画問題 (Vehicle Routing Problem, VRP) と呼ばれる. また, 局所探索法は局所的な改善を繰り返す発見的解法であり, 厳密解を得ることが困難な最適化問題に対し近似解を得る強力な手法である. しかし, 現実の運搬経路問題においては複雑な制約条件を考慮するため, 実行可能領域内で効率的な局所探索を行えないことも多い. 今回, 富士通グループ内の事例研究において, 顧客間の先行順序制約, および車両が合流して分担作業をするための制約条件を考慮した問題に対して, シンプルな局所探索法を提案できたので紹介する. 提案手法は「扱い易い異なる探索空間を用意し, そこから本来の実行可能領域への写像を定義する」方法を用いたものである.

2 事例研究

2.1 問題の概要

問題の主な特徴を挙げる.

- 担当地区優先制約: 顧客の集合 $\mathcal{V} = \{0\} \cup \mathcal{V}_1 \cup \dots \cup \mathcal{V}_\ell$ は “0” で表されるデポ (depot) と複数の地区に分割されており, 各地区には1台以上の運搬車が所属している. 運搬車は所属する担当地区が完了したときにのみ, 他の地区を応援することができる.
- 先行順序制約: 各地区 \mathcal{V}_i ($i = 1, 2, \dots, \ell$) には半順序が定義されており, $i \succ j$ ならば, 顧客 i での作業完了は顧客 j での作業開始よりも前でなければならない.
- 合流可能制約: 複数の運搬車が合流することで, 顧客 i の要求作業時間 w_i を分担して消化することができる. このとき, 途中合流は可能であるが, 作業完了前に離脱したり, 合流せずに異なる時間帯に作業をすることは許されないものとする. 途中合流する場合は, 滞在 (作業) 時間に比例して w_i を消化する. ただし, 合流が許されるのは, 合流によりその顧客の作業完了時刻を 20 分以上改善できるときのみとする.

- 夜間休憩制約：各運搬車は早朝 06:00 にデポを出発することができる。ただし、前日のデポ到着からの時間が8時間に達するまでは再出発することができない。また、夜 22:00 以降は新たな顧客での作業を開始してはならない。22:00 を過ぎると、作業中の顧客での作業が終了次第デポに帰還する。
- min-max 型の目的関数：全体の運搬業務が完了するまでの時間を最小化する。すなわち、デポに戻ってくるのが最も遅い運搬車が帰還するまでの時間を考える。

「合流可能制約」と呼ぶのは、合流巡回セールスマン問題 [3] と区別するためである。合流 TSP は複数のセールスマンが分担して全ての顧客を 1 度ずつ訪問する分担巡回路を求める複数人 TSP (Multiple TSP) の拡張問題であり、2 人のセールスマンが合流して訪問すべき顧客の集合 $\mathcal{V}' (\subset \mathcal{V})$ が所与の問題である。本稿で扱う問題は、合流すべきか否かが決定変数であることを強調しておきたい。

2.2 問題の定式化

ここでは、解きたい問題を正確に伝えるという意味で、問題を数理計画問題として定式化する。「夜間休憩」制約は通常の VRP でよく扱われる時間枠 (time window) とも考えることができ本質的ではない。そこで定式化が煩雑になるのを避けるために本節の定式化および第 3 節の提案手法においては触れないこととする。ただし、後に示す数値結果は「夜間休憩」を考慮したものとなっている。まず、用いる記号を以下に挙げる。

- $\mathcal{V} = \{0, 1, \dots, N\}$ は顧客の集合。“0” はルートの起点となるデポ (depot) を表す。
- $\mathcal{K} = \{1, 2, \dots, M\}$ は利用可能な運搬車の集合。
- \mathcal{V} ：先行順序が定められた顧客対 (i, j) とその先行順序を守らなければならない運搬車 k の 3 つ組 (i, j, k) の集合。すなわち、

$$\begin{aligned}\mathcal{P}' &:= \{(i, j) \mid i \succ j, i, j \in \mathcal{V}\} \times \mathcal{K}, \\ \mathcal{P}'' &:= \{(i, j, k) \mid i \succ_k j, i, j \in \mathcal{V}, k \in \mathcal{K}\}, \\ \mathcal{P} &:= \mathcal{P}' \cup \mathcal{P}''.\end{aligned}$$

ただし、 $i \succ_k j$ は運搬車 k の担当地区に顧客 i は属しているが、顧客 j は属していないことを表している。

- d_{ij} ：顧客 i と顧客 j 間の移動に必要な時間を表す所与の定数。
- w_i ：顧客 i の要求作業時間を表す所与の定数。
- x_{ij}^k ：運搬車 k が枝 (i, j) を通るなら 1、通らないなら 0 の値をとる 0-1 整数変数。
- y_i^k ：運搬車 k が顧客 i を訪問するなら 1、しないなら 0 の値をとる 0-1 整数変数。
- s_i^k ：運搬車 k の顧客 i での作業開始時刻を表す変数。 $y_i^k = 1$ のときに意味を持つ。
- t_i ：顧客 i の作業完了時刻を表す変数。ただし、 $t_0 = 0$ は巡回開始時刻を表す。
- τ ：全体の運搬業務が完了する時刻を表す変数。

上記の記号を用いると、解きたい問題は次の数理計画問題として定式化される。

$$\text{Minimize } \tau \quad (1a)$$

$$\text{subject to } t_i + d_{i0} \leq \tau, \quad \forall i \in \mathcal{V} \setminus \{0\}, \quad (1b)$$

$$y_i^k = \sum_{j \in \mathcal{V}} x_{ij}^k, \quad \forall i \in \mathcal{V}, \quad \forall k \in \mathcal{K}, \quad (1c)$$

$$y_j^k = \sum_{i \in \mathcal{V}} x_{ij}^k, \quad \forall j \in \mathcal{V}, \quad \forall k \in \mathcal{K}, \quad (1d)$$

$$x_{ij}^k(t_i + d_{ij}) \leq s_j^k, \quad \forall i, \forall j \in \mathcal{V}, \quad \forall k \in \mathcal{K}, \quad (1e)$$

$$t_i \leq s_j^k, \quad \forall (i, j, k) \in \mathcal{P}, \quad (1f)$$

$$\sum_{k \in \mathcal{K}} (t_i - s_i^k) y_i^k = w_i, \quad \forall i \in \mathcal{V} \setminus \{0\}, \quad (1g)$$

$$20y_i^k \left(\sum_{g \in \mathcal{K}} y_i^g - 1 \right) \leq t_i - s_i^k, \quad \forall i \in \mathcal{V} \setminus \{0\}, \quad \forall k \in \mathcal{K}, \quad (1h)$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall i, \forall j \in \mathcal{V}, \quad \forall k \in \mathcal{K}, \quad (1i)$$

$$y_i^k \in \{0, 1\}, \quad \forall i \in \mathcal{V}, \quad \forall k \in \mathcal{K}, \quad (1j)$$

$$0 \leq s_i^k < \infty, \quad \forall i \in \mathcal{V}, \quad \forall k \in \mathcal{K}, \quad (1k)$$

$$0 \leq t_i < \infty, \quad \forall i \in \mathcal{V}, \quad (1l)$$

$$0 \leq \tau < \infty. \quad (1m)$$

目的関数 (1a) と制約式 (1b) は運搬業務完了に要する時間は全ての運搬車が作業を終えてデポに戻ってくるまでの時間であることを表している。0-1 変数のみからなる制約式 (1c) と (1d) は各運搬車が各顧客を訪問する回数は高々1回であり、訪問するならば入次数と出次数はともに1、しないならばともに0であることを示している。(1e) は運搬車 k が顧客 i から顧客 j へ直接移動し作業をするときの移動時間を考慮した制約条件である。(1f) は顧客間の先行順序制約および各運搬車の担当地区優先制約である。(1g) は各顧客での作業完了に必要な要求作業時間はそこを訪問する運搬車の作業時間に比例して分担されることを示している。(1h) は合流可能条件であり、「合流することによって、その顧客の作業完了時刻を20分以上改善できる」ときにのみ合流が行われることを表している。その他の制約条件 (1i)~(1m) は各変数が動く領域を表している。

2.3 事例研究での役割分担

制約式 (1e), (1g), および (1h) はいわゆる “Big M ” を用いて、汎用 MIP ソルバーで解くことができる混合整数計画問題 (Mixed Integer Programming Problem: MIP 問題) の形に書き換えることができる。しかし、Big M を含む式が多いこと、問題に TSP の構造を含むこと、目的関数がミニマックス型であること、1つの地区に複数の運搬車が所属する事から生じる解の対称性など MIP ソルバーが苦手とする構造 (宮代・松井 [7] を参照) を多く含んでいることから厳密に解くことは極めて困難である。

図1は富士通グループ内の現場から持ち込まれた顧客数60、運搬車数7のインスタンスに対してエージェントベースシミュレーション環境 “SOARS” を用いてランダムサーチ

を行った結果である．ここでいうランダムサーチとは、TSP に対する最近傍法を乱択化

実行環境：Intel(R) Core(TM) i7 CPU 1.60GHz 4.00GB メモリ，実行時間：56 分

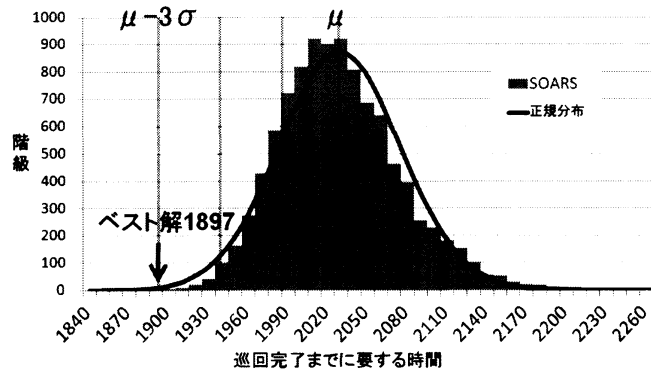


図 1: ABS 環境 SOARS を用いたシミュレーション

したもので、「移動可能な運搬車は制約条件を満たしつつ、近傍内の顧客をランダムに選択する」規則の下で 10,000 回のシミュレーションを行い、その中のベスト解を採用しようとするものである (詳細は吉良・岩根 [2] を参照)．問題の規模が大きくなると効果は薄れていくが、結果を現場のスケジュール計画者に提示したところ、「満足できる解である」という回答が得られた．そこで、ランダムサーチ解に相当する良解を短時間で得るツールの開発が事例研究での課題となり、富士通の開発チームが貪欲的な方法に現場のノウハウを組み入れた構築法を開発し、ランダムサーチ解よりも良い解を数秒の計算で出力している．著者らはさらに良い解を得るべく局所探索法によるアプローチを担当することとなった．

2.4 単純な局所探索法適用時の難点

近年の計算機能力の飛躍的向上もあり、局所探索法は \mathcal{NP} -困難であるような組合せ最適化問題に対して近似解を得る手法として非常に強力である．例えば、複数人 TSP に対しては次のようなシンプルな局所探索法が考えられる．

- Step 0. 顧客の集合を重複なくクラスターに分割： $\mathcal{V} \setminus \{0\} = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_M$ ．

- Step n . これ以上目的関数が改善しなくなるまで以下を繰り返す．

$\max_k t(\{0\} \cup \mathcal{S}_k)$ が改善するように \mathcal{S}_i から \mathcal{S}_j への 1 点挿入、 \mathcal{S}_i と \mathcal{S}_j の 2 点交換等により分割を更新．ただし、 $t(\{0\} \cup \mathcal{S}_k)$ はデポと \mathcal{S}_k 全点の (近似) 最短巡回路長で、Lin-Kernighan アルゴリズム [6] など TSP に対する局所探索法を用いて計算．

複数人 TSP の拡張問題である VRP に対してもこのような方法を上手く適用できれば、応用に十分耐えることができると言われている．確かに、「合流」という概念に対処するには、分割を重複を許す形に変え、運搬車を適当に選んで他の運搬車との合流が実現するようにルート of の適切な位置に顧客を挿入する近傍操作や、逆に合流が解消されるようにルートから顧客を削除する近傍操作等が考えられる．しかし、著者らが試行錯誤した結果、良い結果を得ることができなかった．その理由の一つとして、近傍を探索する際に先行順序

制約および合流可能制約が満たされなくなる可能性が極めて高いということが挙げられる．図2は顧客数9, 運搬車数2の問題例(表1)に対する実行不可能解の一例である．運搬車1のルートだけに注目すると先行順序制約にも担当地区優先制約にも違反していない．運搬車2についても同様である．しかし, 全体としては先行順序制約及び合流可能制約が満たされていない．また, Case 1については, ルートに待機時間を挿入することで実行可能な解に修正可能であるが, Case 2では待機時間挿入による修正は不可能である．

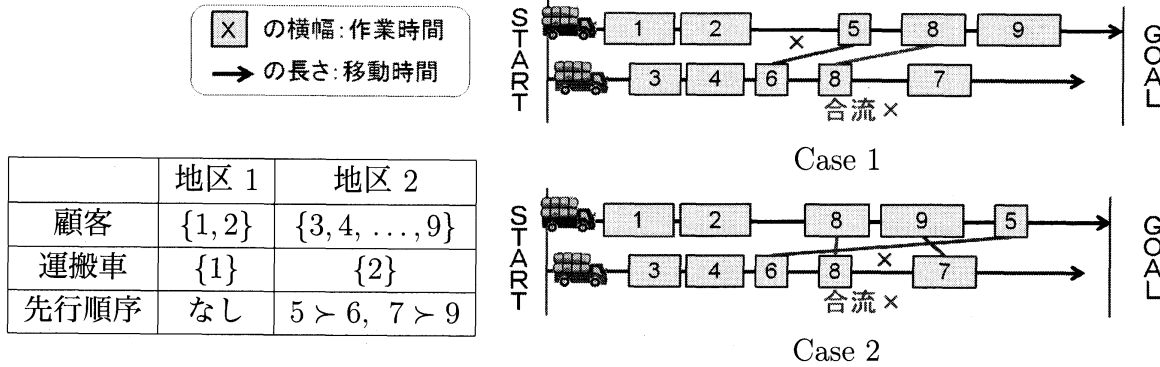


表 1: 顧客数 9, 運搬車数 2 の問題

図 2: 例題 (表 1) に対する実行不可能解の例

3 提案手法

実行可能領域において局所探索を効率的に行うことが難しい場合, 異なる探索空間を用意し, その中で局所探索を行う方法が提案されている (図3参照). 探索空間から実行可能領域への写像を介して解の評価を行うため, 探索に時間はかかるという欠点があるが, 制約条件が複雑なスケジューリング問題など, 問題によっては非常に有効であることが知られている [1, 8]. この考え方に基づいて, 先行順序付き合流可能 VRP に対する局所探索アルゴリズムを設計する.

3.1 探索空間と実行可能領域への写像

顧客 $1, 2, \dots, M$ の置換の全体を \mathfrak{S}_M とし, $\mathfrak{S}_M^{(k)}$ ($\subset \mathfrak{S}_M$) を先行順序制約及び運搬車 k の担当地区優先制約を満たす置換全体とする. また, 探索空間を $\mathcal{X} := \prod_{k \in K} \mathfrak{S}_M^{(k)}$ と定義する. ここで, 探索点 $x \in \mathcal{X}$ を M 行 N 列の行列として表記することにする. 表1の例題に対しては, 例えば $x \in \mathcal{X}$ として, 以下が挙げられる.

$$x = \begin{pmatrix} \sigma_1 \\ \sigma_2 \end{pmatrix} = \begin{pmatrix} \sigma_1(1) & \sigma_1(2) & \cdots & \sigma_1(9) \\ \sigma_2(1) & \sigma_2(2) & \cdots & \sigma_2(9) \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 5 & 7 & 6 & 8 & 9 & 4 \\ 3 & 7 & 5 & 6 & 8 & 4 & 9 & 1 & 2 \end{pmatrix} \quad (2)$$

いま, 2つの先行順序制約 $5 > 6$, $7 > 9$ が与えられているので, どちらの行も5は6より左に, 7は9より左にある. また, 第 k 行は運搬車 k の担当地区優先制約を満たさなければならないから, 第1行では $1, 2$ は $3, 4, \dots, 9$ よりも左にあり, 逆に第2行では右にある.

この探索点 x に対して、図4のように実行可能解を対応させる写像 f を考える。すなわち、運搬車 k はリスト σ_k に従って、 $\sigma_k(1), \sigma_k(2), \dots, \sigma_k(N)$ という順番で全顧客を訪問しようとする。このとき、顧客 $\sigma_k(i)$ への「移動を開始する時点」で「顧客 $\sigma_k(i)$ への移動を開始するのは運搬車 k が最初である」、もしくは「既に他の運搬車が作業中あるいは移動中であるが、今から移動を開始すれば合流可能条件 (20 分以上の作業時間の短縮に寄与) が満たされる」場合にのみ実際に訪問する。それ以外の場合はその顧客を飛ばし、次の顧客 $\sigma_k(i+1)$ を参照する。図5は探索点から実行可能解を生成し、目的関数値を計算するた

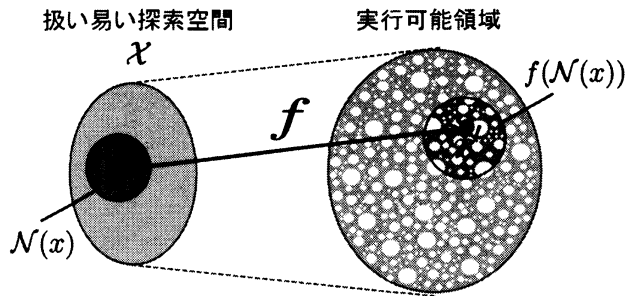


図3: 異なる探索空間を用いる方法

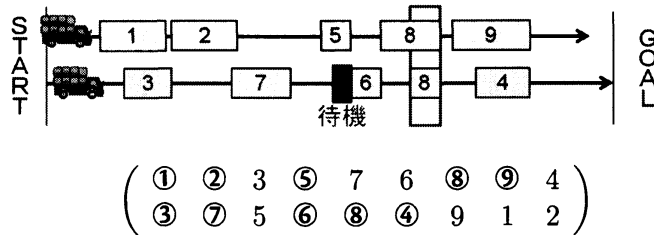


図4: 探索点 x に対応する実行可能解 $f(x)$

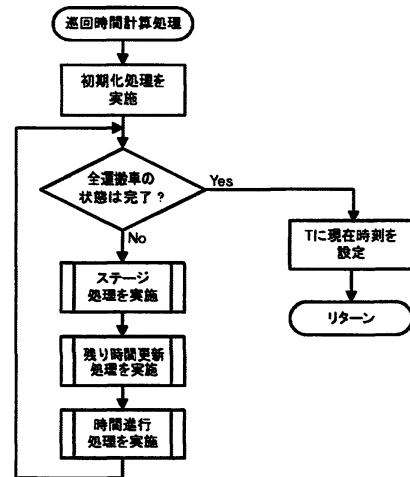


図5: 写像 f による評価

めの具体的な手順を示しており、3つのサブルーチンからなる。各サブルーチンの具体的な手順は図6、図7および図8に示している。また、フローチャートで用いる運搬車の属性値を表す変数を表2に示す。

変数名	概要
状態	初期 : 業務開始前 移動中 : 顧客あるいはデポへの移動を行っている状態 作業中 : 顧客上で作業を行っている状態 待機中 : 顧客に到着したが制約条件から作業を開始できない状態 完了 : 運搬業務が終了しデポに帰還した状態
位置	「状態 = 移動中」のときは行き先、それ以外は運搬車が現在いる場所
残り時間	現在の移動または作業が完了するまでの時間

表2: フローチャートで用いる運搬車に関する用語

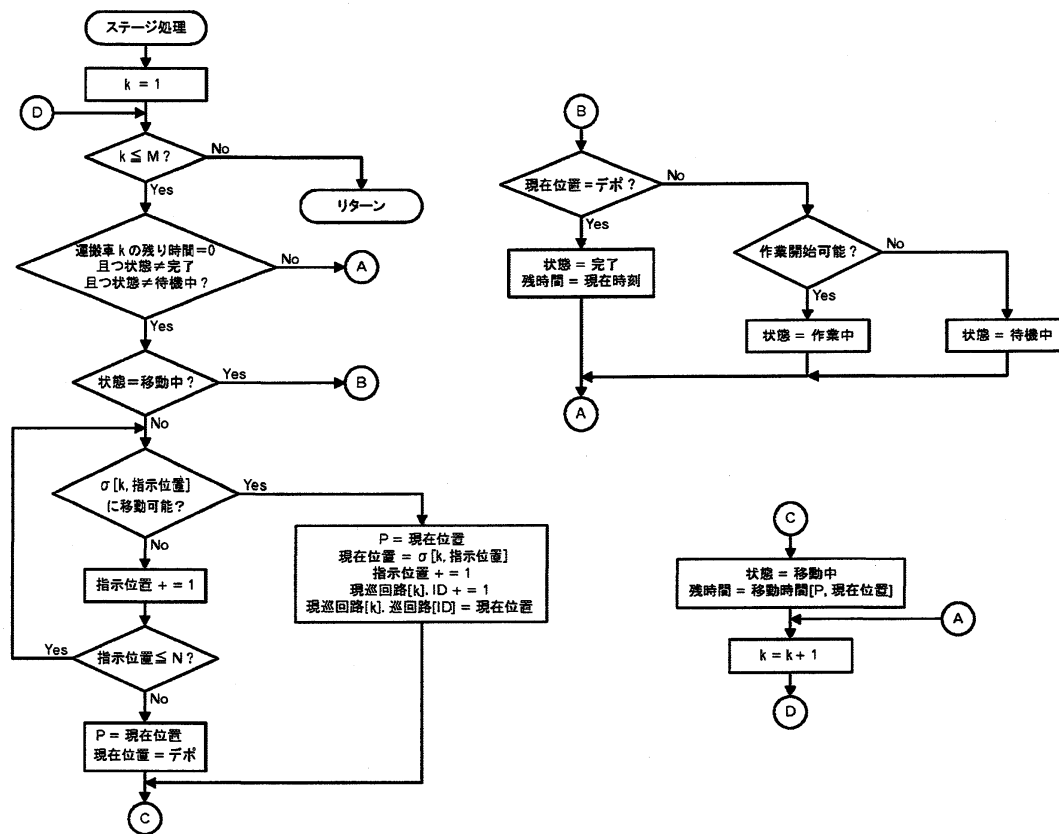


図 6: ステージ処理

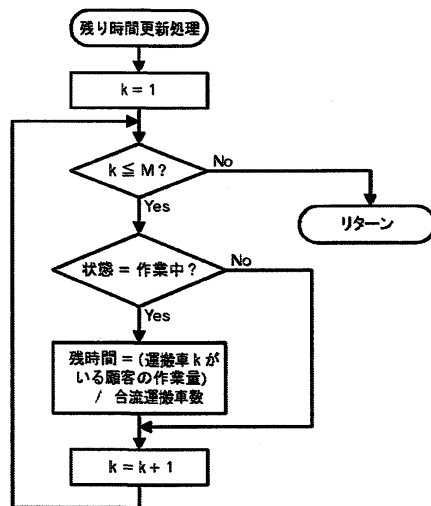


図 7: 残り時間更新処理

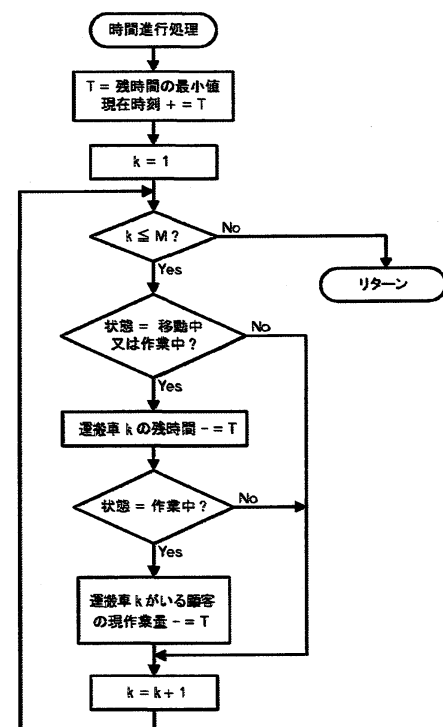


図 8: 時間進行処理

3.2 探索近傍と局所探索

探索近傍 $\mathcal{N}: \mathcal{X} \rightarrow 2^{\mathcal{X}}$ を x の intra-swap 近傍とする。すなわち,

$$\mathcal{N}(x) := \bigcup_{k \in \mathcal{K}} \{(\sigma_{-k}, \sigma'_k) \mid \sigma'_k \text{ は } \sigma_k \text{ を 1 回 互換 した もの, } \sigma'_k \in \mathfrak{S}_M^{(k)}\},$$

$$x = (\sigma_1, \sigma_2, \dots, \sigma_M) \in \mathcal{X}.$$

ただし, $(\sigma_{-k}, \sigma'_k) := (\sigma_1, \dots, \sigma_{k-1}, \sigma'_k, \sigma_{k+1}, \dots, \sigma_N)$. 図 9 は例題 (表 1) に対して, (2) 式の $\sigma_2(2)$ と $\sigma_2(6)$ を入れ替え写像 f で評価した結果として目的関数が改善した様子を示している. 図 10 は局所探索を行う手順を示している. 第 2.4 節の図 2 において, 各運搬車のルートだけに注目したときに先行順序制約および担当地区優先制約が満たされているとしても, 全体としてはそれらの制約が満たされていない可能性があることを指摘した. しかし, 第 3.1 節で定義した写像を用いると, 各運搬車のリスト σ_k が制約条件を満たす (すなわち $\sigma_k \in \mathfrak{S}_M^{(k)}$) ならば $x = (\sigma_1, \sigma_2, \dots, \sigma_M)$ から必ず実行可能解が生成される. したがって, intra-swap が実行可能か否かを局所的な情報から確認できる.

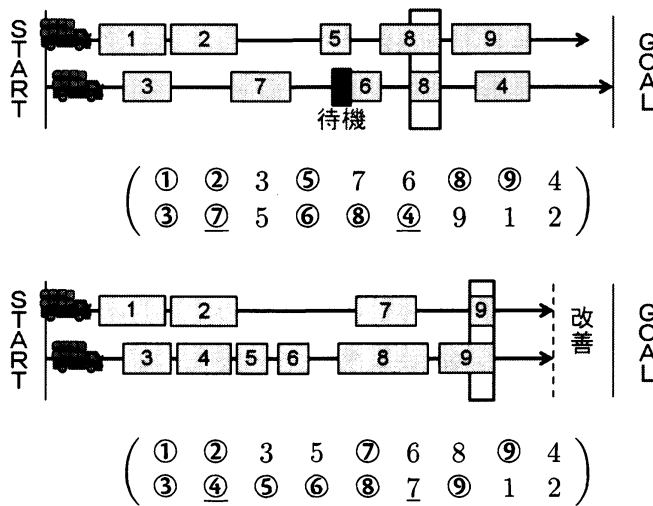


図 9: intra-swap による解の局所改善の様子

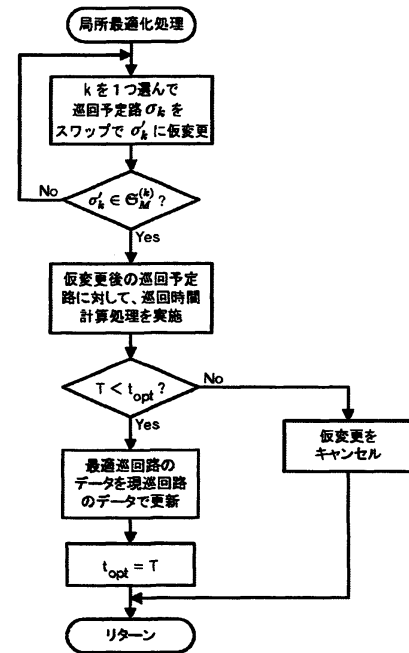


図 10: 局所探索処理

3.3 提案手法による効果

図 11 は事例研究における顧客数 60, 運搬車数 7 のインスタンスに対して, ランダムに 3 つの初期解を生成し, 第 3.2 節で定義した近傍に基づいて反復局所探索を行った結果である. 実装は C 言語を用いて行っている. 図 12 にアルゴリズム全体の流れを示す. 早いものでは約 14 秒で構築法の解を突破している. 最終解が最も良かった B においては, 夜間休憩 8 時間を除くと, 構築法の解よりも約 8 % 目的関数を改善したことになる. ゆえに, 提案した局所探索法を用いた多スタート戦略により, 十分に良い解が得られると期待できる.

例えば、複数人 TSP においては(局所探索法の観点からは)難しい制約条件はなく、簡単に局所探索法を設計できる。しかし、1つの近傍操作だけではなく、複数の近傍操作を上手く組み合わせて用いなければ効果が発揮されないことが知られている。我々のアプローチにおいては数値実験から得られた観察として、探索空間 \mathcal{X} の中では intra-swap という単純な近傍操作しか考えていないにも関わらず、写像によって生成される実行可能解はごく局所的な変形だけでなく、図9のように大幅な変形も行われている。探索空間と写像のこのような性質が良い数値結果に結び付いたのではないかと著者らは考えている。

実行環境：Intel(R) Xeon(R) CPU X5570 2.93GHz 11.7 GB メモリ

	初期解	最終解	最終更新時の計算時間	構築法の解突破時の計算時間
A	2259	1772	7分01秒	14秒
B	2182	1760	13分06秒	2分12秒
C	2088	1863	24分36秒	15分07秒

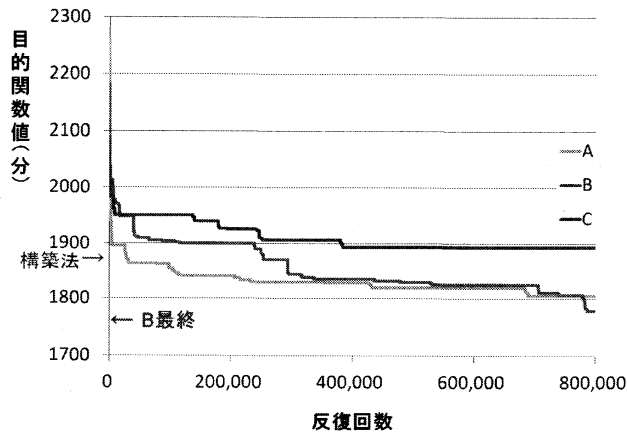


図 11: 反復局所探索による解の改善

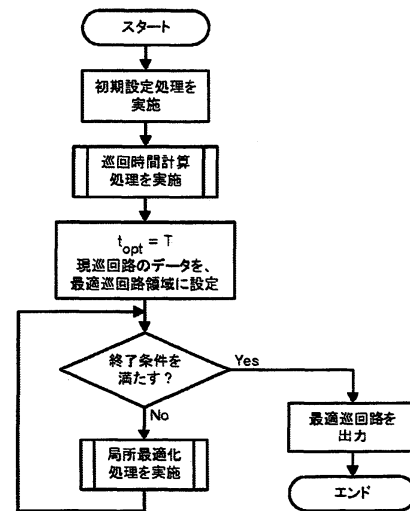


図 12: 全体の流れ

4 まとめと今後の課題

先行順序制約, および合流可能制約という難しい制約条件をもつ運搬経路問題に対して, シンプルな局所探索法を提案した. 提案手法は「扱いやすい探索空間を用意し, 本来の実行可能領域への写像を定義する方法」の応用例である. 今後は問題の規模が大きくなったときの影響を分析する予定である. また, 探索空間内での近傍の定義, 走査順序, 移動戦略については工夫の余地があり, これらの検討は今後の課題としたい.

謝辞

多くの有益な助言やコメントを頂きました大阪大学の梅谷俊治先生に御礼申し上げます.

参考文献

- [1] 今堀慎治, OR 辞典 Wiki「ハイブリッドメタ戦略」, オペレーションズ・リサーチ「経営の科学」, 2009 年 12 月号, pp.72-73.
- [2] 吉良知文・岩根秀直, エージェントベースシミュレーションを用いたある運搬経路問題の解析, 計測自動制御学会 第 44 回システム工学部会研究会予稿集 (PDF), 2011 年 3 月, pp.51-54.
- [3] 小林克也・沼田一道, 合流巡回セールスマン問題とその解法, オペレーションズ・リサーチ「経営の科学」, 2011 年 1 月号, pp.33-39.
- [4] G. Laporte & I. H. Osman, Routing Problems: A Bibliography, *Annals of Operations Research*, **61** (1995), 227-262.
- [5] J. K. Lenstra & A. H. G. Rinnooy-Kan, Complexity of Vehicle Routing and Scheduling Problems, *Networks*, **11** (1981), 221-227.
- [6] S. Lin & B.W. Kernighan, An effective heuristic algorithm for the traveling salesman problem, *Operations Research*, **11** (1973), 498-516.
- [7] 宮代隆平・松井知己, ここまで解ける整数計画, システム/制御/情報, **50** (2006), pp.363-368.
- [8] 柳浦睦憲・茨木俊秀, 組合せ最適化—メタ戦略を中心として—, 2001 年, 朝倉書店.
- [9] 山本 芳嗣・久保幹雄, 巡回セールスマン問題への招待, 朝倉書店 (1997).